7th International Command and Control Research and Technology Symposium
Monitoring the Command and Control Weapon System
A Master Caution Panel for C2 System Monitoring

Author Name:

Martin J. Brown Jr
BBN Technologies
9655 Granite Ridge Drive, Suite 245
San Diego, CA 92123

Abstract

This paper will highlight the Air Force Research Laboratory (AFRL) sponsored research being carried out by a team led by BBN Technologies to develop a Master Caution Panel (MCP) for modern, distributed command and control systems. The primary purpose of MCP is to provide military decision makers with situational awareness of the command and control weapon system. MCP accomplishes this by taking information technology (IT) resource data in the form of IT events and transforming them into operationally significant information. MCP bridges the gap between operational personnel and systems support personnel by providing each group with an independent views of the same set of IT related events. The operator can easily understand the impact of IT resource problems on mission accomplishment and can provide the systems support personnel with clear prioritization concerning where emphasis on fixing problems will have the most impact.

1 Introduction

This paper highlights Air Force Research Laboratory (AFRL) sponsored research being carried out by a team led by BBN Technologies to develop a Master Caution Panel (MCP) for modern, distributed command and control systems. The primary purpose of MCP is to provide military decision makers with situational awareness of the command and control weapon system. MCP accomplishes this by taking information technology (IT) resource data in the form of IT events and transforming them into operationally significant information. MCP bridges the gap between operational personnel and systems support personnel by providing each group with an independent views of the same set of IT related events. The operator can easily understand the impact of IT resource problems on mission accomplishment and can provide the systems support personnel with clear prioritization concerning where emphasis on fixing problems will have the most impact.

Specifically, this paper examines MCP's use of Sun Microsystems' Jini technology to provide a distributed monitoring environment that supports multiple independent views of system status

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
SEP 2002		**00-00-2002 to 00-00-2002**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Monitoring the Command and Control Weapon System. A Master Caution Panel for C2 System Monitoring	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **BBN Technologies,9655 Granite Ridge Drive Suite 245,San Diego,CA,92123**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**16**	

through a decoupling of monitors and client applications. MCP employs Jini's registration, discovery, inquiry and service contract mechanisms to allow operators to obtain views that are linked to specific mission related concerns. The operator specific views and the linkage of resources to functional tasks are maintained in a MCP knowledge base. The paper addresses methodologies to filter the vast number of IT resource events into a meaningful set if IT resource status information. It discusses the use of mobile code as part of the MCP service contract mechanism and the potential uses of mobile code to implement remedies for system deficiencies.

The Master Caution Panel allows military decision makers to focus on the critical, time sensitive decisions that directly impact the lives of the men and women charged with executing those decisions. MCP provides these decision makers with clear, unambiguous status information regarding those information technology systems that support the command and control process.

2 The Master Caution Panel for The Command and Control Weapon System

Dramatic improvements in the delivery speed and accuracy of modern weapons place greater demands on military decision makers to process and act upon a constantly changing situation. To support this decision process complex, modern command and control systems have evolved as a true system of systems. Specialized components, developed independently are joined together, often through a "forced marriage" of interoperability.

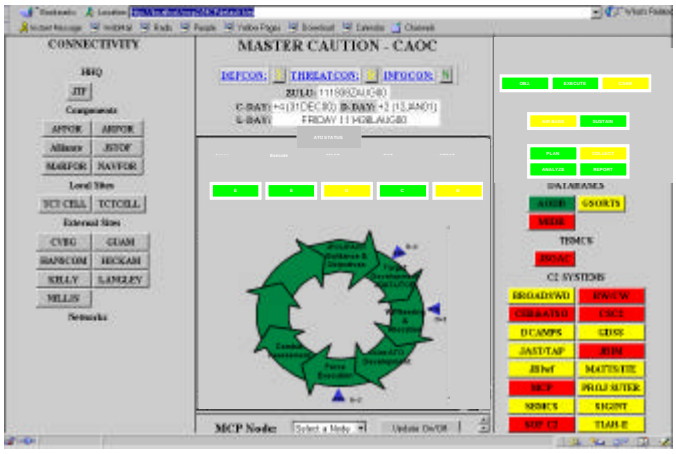

Figure 1 Master Caution Panel Display

The result is a system that, although is highly effective operationally, is poorly understood by both operators and technical support personnel in the field. Critical indicators of potential problems are usually missed. Corrective action is normally taken only as a result of a catastrophic failure and consists of shutting down the affected system and rebooting. This response costs time in an environment where the difference in success and failure is measured in minutes and seconds and lives and valuable equipment is put at risk.

The current situation is unacceptable. As the Department of Defense moves toward increased reliance on distributed command and control as envisioned by Network Centric Warfare (NCW) there will be demands for increasingly complex technology implementations to support distributed, cooperating command and control. What is often overlooked during the design and

development of these modern systems is the fact that these NCW concepts also necessitate a new and revolutionary concept for monitoring distributed enterprise systems.

The concepts of Network Centric Warfare and Global Information Grid (GIG) represent revolutionary approaches to military command and control. Accompanying the adoption of these concepts is a growing realization that the facilities and systems supporting command and control are a weapon system in every sense of the term. Like any other weapon system, the trained personnel operating the system require timely, accurate and meaningful information regarding the status of the weapon system. In military aircraft, a Master Caution Panel alerts the pilot to potential problems with critical aircraft systems. This is a key factor that bears repeating. The Master Caution Panel is an alert system. It does not provide any status information on aircraft systems. Instead, it alerts the pilot when specific indicators cross pre-determined thresholds. It allows the pilot to concentrate on the mission and to treat aircraft system problems by exception. This concept forms the cornerstone for the Command and Control Master Caution Panel.

Senior DoD decision makers have called for automated monitoring and management tools to keep essential command and control systems operating. Commercial tools, while providing partial solutions, cannot provide the comprehensive end-to-end capability needed. In several instances, the proposed solution has been to combine two or more specialized commercial products. Combining multiple COTS products adds breadth to the coverage but results in a monitoring and management system that potentially exceeds the complexity of the C2 systems being monitored.

The long-term need is for modern systems that are self-configuring and self-healing. These systems would incorporate technologies to recognize potential problems, diagnose the cause and impact, advise the operators, and implement remedy actions either with or without explicit permission of the system administrator.

Figure 2 highlights several important issues in monitoring a distributed enterprise.[1] Items one, two, and three are obvious but often overlooked because they present significant difficulties to the developers of distributed monitoring systems. Item four and five address the issue of perspective in determining the health of system resources. Perspective here means that users at different locations may have significantly different beliefs concerning the availability of specific system resources. Part of this difference may be attributable to differences in observer location in the network topology.

---

[1] BBN Distributed Monitoring Architecture (BDMA) Description. BBN Technologies, June 2001,

Another more significant difference arises due to different perspectives regarding the specific criteria used to determine health. The normal response from a system perspective is to attempt to standardize on a specific set of criteria.

The true question is, "If multiple assessment criteria can be used to determine system status, then is the monitoring system must be capable of supporting these divergent needs?"

Item six reflects the barrier that often separates IT personnel and functional users. Users are more concerned with the operational impact of system problems than they are with the specific problem. The monitoring system needs to translate detected information technology events into meaningful operational impacts.

Item seven argues against an imbedded monitoring system while item eight cautions against a system that increases demands on limited system resources in response to perceived problems. The monitoring system should run in parallel with the system being monitored but not place a significant burden on already taxed IT resources.

Issues in Enterprise System Monitoring

1. *The status depends on the health of a large number of system resources.*
2. *Resources can be widely distributed both geographically and topologically.*
3. *Enterprise systems can be composed of subsystems, and the decomposition of resources is not necessarily hierarchical*
4. *The observed status of a resource is highly dependent on where the point from which the status is being monitored.*
5. *Different users have different notions of what "healthy" means, and even what "system" and "enterprise" mean.*
6. *Users of an enterprise system are less concerned with the specific cause of a fault than they are with the impact of the fault on their workflow.*
7. *Any system to monitor enterprise system resources must be more resistant to failure than the resources being monitored.*
8. *A system that monitors the status of an enterprise system in order to provide early warning and diagnosis of problems must not become a contributing factor to these problems.*

Figure 2: Issues in Enterprise Monitoring

3 Resources and Monitors

Before discussing specific elements of the Master Caution Panel design and implementation it is important to define the terms resources and monitors. In the MCP context, the command and control system is supported by a series of resources. MCP's definition of a resource may be slightly different than the definition of a resource (even the same resource) from a different perspective. But it's important in the development of an MCP monitoring system to consider potential MCP entities from a monitoring perspective.

3.1 Resources

A *resource* is anything that has a status and can be identified by its metadata. A resource can be concrete, virtual, ephemeral, or even conceptual to name only a few. The status information of a resource must be able to be exported via a well-defined interface. Resource event reporting in MCP follows the industry standards of SNMP and WBEM. We can now provide some example resources along with descriptions of their status to make things more clear.

3.1.1 Concrete Resources

A network router is concrete for several reasons. One is that it's physical. You could put a property tag on it. Another is that performs some well-defined function. A database application may not be physical in the same sense as a router (we wouldn't put a property tag on the bits on disk), but it is concrete because the application has a well-defined behavior in the way that it acts upon enterprise data. The status of a router can be represented by throughput level or the number of simultaneous connections is supports. Typically, a router would also have specifications, and while these aren't directly part of its status, they provide a baseline for status metrics, like current throughput as a fraction of rated capacity. The router may also have an IP address. This is probably not status information, but rather is part of its metadata.

3.1.2 Virtual Resource:

Wireless data link: A wireless data link is virtual because it only exists when you need it. Its status could be defined in terms of the bit rate, the bit error rate, the bit rate as a fraction of the link capacity, etc.

3.1.3 Ephemeral Resource:

A task: A task is ephemeral because it only exists until it is done. If our router breaks, we buy a new one to replace the broken one. If a task completes, we (usually) don't find another one to replace it. The status of the task may be its percent completion, or the fraction of budget spent, etc.

3.1.4 Conceptual Resource:

A theory: The status of a theory could be the level of acceptance in the community of interest, or its ability to explain existing phenomena, or the degree to which it can be verified by experiment.

3.2 Monitors

Monitors, in the Master Caution Panel context represent the foundation of the alerting capability. MCP monitors are relatively simple components that are decoupled from the client applications that are consumers of monitor events. A monitor consists of an event handler that captures resource events from an event source; the monitor body that transforms the resource events into MCP events and provides all Jini related interactions with the registry; an administrative

interface that allows the monitor owner to configure and activate the monitor and, a consumer or public interface that provides two critical functions. First, the consumer interface provides the interface that consumers use to receive the property/value pairs. Second, the consumer interface supports inquiry by potential consumers responding with the monitor metadata and a complete list of the monitor-exposed properties. Finally, the consumer interface provides execution support of the mobile code instructions that are part of a service contract.

3.2.1 Monitor Types

The Master Caution Panel employs a number of both simple and complex monitors as shown in Figure 3.

Monitor	Description
Latency Monitor	This is the simplest of the MCP monitor family. The latency monitor employs a simple "ping" to determine if a physical resource is reachable and the latency in round trip time for a message to reach the resource and the reply to return. The primary properties exposed by the Latency Monitor are health with the values of "ALIVE", "DEAD", and latency with a number reflecting the round trip time.
SNMP Host Monitor	The SNMP Host Monitor is used to expose properties of a physical host. The SNMP Host monitor can expose host related properties such as CPU utilization, Memory Usage, and the Process table.
SNMP Application Monitor	The SNMP Application Monitor can expose relevant information concerning a single software application running on a host. The monitor can expose application specific information such as CPU utilization and Memory Usage
SQL Monitor	The SQL Monitor is capable of repeatedly executing a single query against a database. The monitor contains a simple set of business logic to transform the specific database values into MCP relevant property/value pairs.
Non-IP Device Monitor	MCP can monitor non-IP devices such as Uninterruptible Power Supplies (UPS) and telephone switches (when an interface card with a defined API is installed)
HPOpenview Monitor	The HPOpenview monitor is used to demonstrate the process of obtaining specific resource information from a third party system management application. HPOpenview was selected because it made a public API available and is used in most US military network operations centers
Business Logic Monitor	The Business Logic Monitor is a complex monitor often establishing service contracts with one or more simple monitors then fusing and transforming the property/value pairs obtained from these monitors into relevant information about a process, product or conceptual resource.

Figure 3: MCP Monitor Types

3.2.2 Monitor Resource Relationship

The "monitor" monitors exactly one resource. It interprets the state of the resource in terms of "properties", which are simply a name, a type, a value, and a set of "navigators". It exposes these properties to BDMA clients. As we will see, this one-to-one relationship between resources and monitors is important. When one considers deploying a monitor, the first and most important questions that need to be answered are:

- What resource is being monitored?
- What are the status properties of that resource?
- What kinds of values do those status properties hold?

4 Master Caution Panel Architectural Framework

To satisfy the identified requirements for the Air Operations Center Master Caution Panel a service-based architecture was selected. This type of architecture has a provider-consumer relationship as its foundation. Master Caution Panel monitors are producers of resource status information. An application (User Visualization or other monitor) that is interested in the type of resource information can find the service providers that support its information needs through a process called “Discovery”. Once discovered, the application and the monitors enter into a service contract(s) through a negotiation process. The contract is completed and the client uses Java’s mobile code capabilities to send specific processing instructions to the service provider. This discovery and contract process facilitates a decoupling of the monitors and their clients.

Decoupling provides enhanced flexibility for MCP to satisfy the diverse requirements that different user groups have for system resource status information. A monitor can be built with the focus on the monitored resource and the basic elements of information that provide indications of resource health. The monitor has no knowledge of the information needs of the various clients it enters into contracts with. A single monitor is able to satisfy the information requirements of several clients. Figure 4 demonstrates this concept for an SNMP Host monitor that reports CPU usage as one of its properties. In the example below, three simple monitors are monitoring a resource. An aggregate monitor has entered into a contract with each of these monitors for specific information elements. Additionally, there are two clients with an interest in the status of the resource. The first client, representing a system administrator enters into a contract with one of the simple monitors. This contract requests detailed information related to resource status. The monitor provides current CPU usage as one of its properties. The first client, established by a system administrator user, provides processing instructions as part of its contract that tell the monitor to “run this routine” each time CPU usage is reported. The routine computes a rolling average over the previous five minutes. The contract establishes a threshold

of 90% for this computed average. It also establishes a threshold of 98% for any individual reading. The contract code provides events to the client when the thresholds are crossed.

A second client representing the aggregate monitor also establishes a contract with the monitor. This contract also computes a rolling average but the time period is ten minutes and the threshold is set at 95% for two consecutive average computations. Again, the contract code determines when events are sent to its respective client.

A functionally oriented user such as the Chief of Combat Plans has established a requirement for information related to this resource. The client that satisfies this need enters into a contract with the aggregate monitor for the relevant information.

The example above demonstrates how a single monitor can support the divergent needs of two or more clients without any *a priori* knowledge of the information processing needs of the clients. The monitor understands how to collect a specific set of resource data elements. Through the contract it agrees to report a subset of these elements to the client contract code. The monitor does not know or care what happens to the data elements after they are provided to the contract code. A monitor obtains status information for a resource and reports this information. The interpretation of this information is left to the clients or the consumers of the information.

The mobile contract code is owned by the client but operated by the monitor. The monitor, as part of the

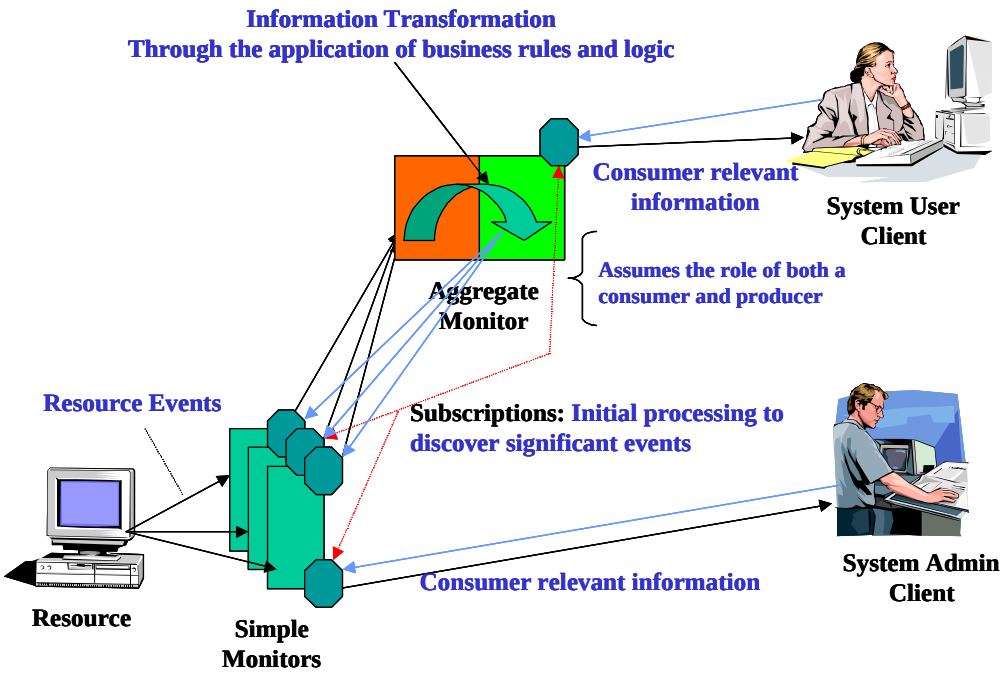

Figure 4: Monitors support multiple clients

contract, agrees to "run the client provided routine" each time a specific set of data elements are reported or at a set time interval.

5 Technical Foundation of the MCP Service-Based Architecture

The MCP implementation of a service-based architecture uses Sun Microsystems' Jini technology as a foundation. Specifically, MCP uses the Jini discovery, look-up and leasing services. BBN Technologies' Distributed Monitoring Architecture (BDMA) extends and enhances the core Jini technologies specifically to support a distributed monitoring environment. The specific MCP implementation of these technologies is described below.

Figure 5 provides a conceptual view of the MCP Architecture and those external entities that have a direct impact on MCP operations. Air Operations Center IT resources are shown at the bottom of the diagram. As stated earlier, these resources are the hardware, software, data and

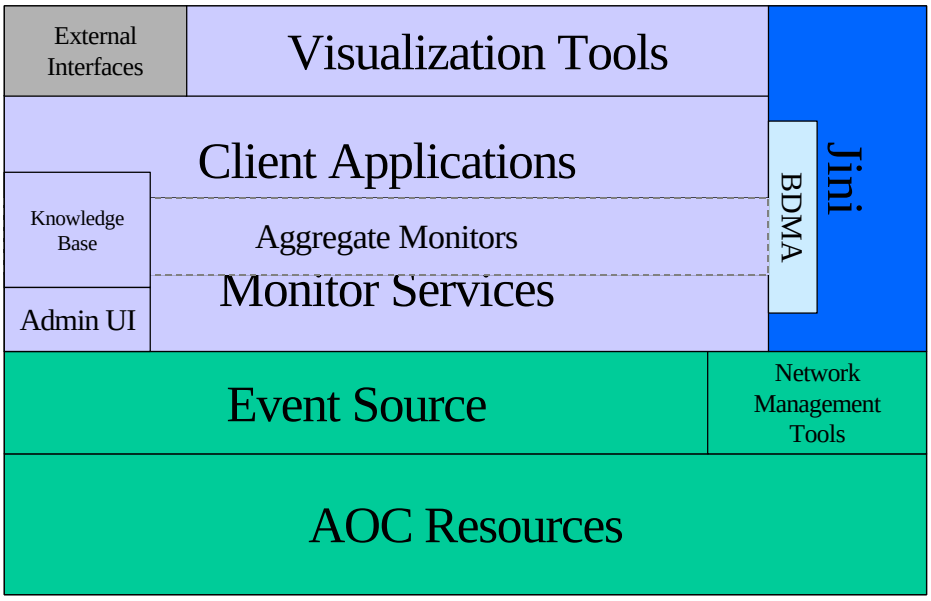

Figure 5: MCP Architecture

communications media that support the air operations mission.

Above the actual resources we show the event source layer. Typically, IT resources are in constant operation and the various properties that reflect status are in a constant state of flux. The key element for a remote monitoring system is to capture these internal property transitions. This is accomplished through the use of the event source layer. This layer reflects the fact that MCP employs an industry standard event reporting structure. MCP receives event from Simple Network Management Protocol (SNMP) Agents and Web Based Enterprise Management (WBEM) CIM Object Manager (CIMOM) sources. This reliance on industry standard resource event reporting protocols reflects a conscious decision to ensure that MCP implementation did not impose a development burden (in terms of implementing a unique reporting interface) and to make MCP monitoring compatible with evolving COTS and GOTS applications. At the right of this Event Source layer the diagram shows an available interface to commercially available

network management tools currently used in an AOC. MCP had developed interfaces to these tools where there is a well-defined published interface and well-defined data representations such as HP Openview. The interface with these commercially developed systems monitoring applications is possible only when the commercial application has a published API and the internal data representations are known.

Directly above the Network Management Tools the diagram shows the Jini services layer. Jini is a freely distributed product of Sun Microsystems. MCP employs Jini look-up and discovery services to support the decoupling of monitors and clients.

The BBN Distributed Monitoring Architecture (BDMA) provides a second external set of capabilities. BDMA provides the basic characteristics and methods for monitors in a distributed environment. A more complete discussion of BDMA is provided later in this document.

Monitoring Services comprises the first true MCP layer. This layer consists of the various monitors implemented within the MCP environment. These monitors take the form of Jini services made available to various clients through the look-up and discovery process. The Monitoring Services also contains a small administrative user interface to support the configuration, deployment, and troubleshooting of individual monitors.

Directly above the Monitoring Services layer is the Client Application Layer. This layer contains the clients of monitoring services. These clients use Jini look-up and discovery to find appropriate monitors. The clients enter into service contracts with the monitors and manage their leases through a renewal process. Clients also employ "business logic" drawn from the knowledge base to transform monitor events into useful consumer information.

Business Logic Monitors share the characteristics of both client applications and monitor services. Aggregates discover and enter into service contracts with other monitors using the same methods as other clients. These aggregates transform the monitor events into process/organization specific status through the use of "business logic" and then make this information available in the same manner as other monitor services.

The top layer reflects the visualizations provided to the various MCP user groups and the interfaces to external information presentation and dissemination systems such as the Joint Battlespace Infosphere (JBI).

The MCP knowledge base is shown overlaying both the monitor services and the client application layers. The knowledge base provides a central repository for information relating specific IT resources to air operations processes and AOC organizational elements. The knowledge base also reflects the business logic used in the process of transforming data into useful information.

5.1 Jini Distributed Technology Services

Jini is a distributed system technology developed by Sun Microsystems. The underlying original concept for Jini reflects a shift from an environment where individual autonomous devices are networked to an environment where these devices join together to form a true distributed system[2]. BBN Technologies was one of the first independent developers to take Sun's device centric concept and apply it to a distributed software services environment.

Jini's ability to support communities of services is based around five key concepts[3]. The distributed monitoring environment of MCP builds on these concepts, therefore an understanding is important as part of the design description. MCP is a Jini service community with the individual monitors being Jini services.

- Discovery: Discovery is the process used to find communities on the network and join with them. MCP uses the discovery methods provided by Jini to support service consumers in finding appropriate service providers.

- Lookup: Lookup governs how the code that is needed to use a particular service finds its way into participants who want to use the service. Jini's lookup services are more complex than a name server allowing complex searches involving Java object types and inheritance relationships.

- Leasing: Leasing is the technique that provides Jini's self-healing nature. It ensures that the community will recover from the loss of any key services.

- Remote events: Remote events are the paradigm Jini uses to allow services to notify each other of changes in their state. Because lookup itself is a service, it can use remote events to notify interested parties when the set of services available to a community has changed.

- Transactions: Transactions are Jini's mechanism for allowing computations that may involve multiple services to reach a safe state. Jini's transaction model helps guard against the problems associated with partial failures in distributed systems and can provide services with robustness and resilience to network failures.

---

[2] Jini Architectural Overview, p 1

[3] Jini Architectural Overview pp 5-8

5.1.1 Advantages of the Jini Service-Based Approach

The primary motivation in the selection of Jini as the foundation of the Master Caution Panel was its support for the dynamic formation of service federations across a distributed computing environment. The Jini Look-Up Service represented a well-defined methodology for consumers to dynamically discover potentially relevant services in the same manner that a person finds a specific type of service using the "Yellow Pages". Neither the person using the phone nor the MCP consumer knows ahead of time the name of the service provider they will establish a relationship with. All that is needed is an understanding of the type of service needed to allow negotiation among service providers and consumers to take place.

Jini also supports communication between service providers and consumers through the Java Remote Method Invocation (RMI™). RMI allows full objects (data, code, methods) to be passed around a distributed enterprise. MCP employs this mobile code capability to provide information transformation and filtering at the source thus limiting MCP use of network bandwidth resources.

Jini also provided important security capabilities. The security model for Jini technology is build on the twin notions of a principal and an access control list.[4] Jini services are accessed on behalf of some entity – the principal which traces back to a particular user of the system. Access to services is regulated through the access control list.

Jini provided clear advantages in these important areas over Corba and other distributed systems.

5.2 Lookup, Discovery and Service Delivery

The process begins with the initialization of Jini Look-up service by the system administrator. The administrator then begins to configure and deploy MCP monitors. The typical employment scenario described in the following paragraphs.

5.2.1 Jini Look-Up Service and Discovery

At initialization, MCP monitors announce their availability and register with the one or more Jini look-up services. The monitors employ a "well defined" set of information to advertise its availability to support user requirements. This information includes metadata about the monitor itself (type, name, etc.) and MCP specific information about the resource it monitors. This domain specific, command and control ontology is maintained by the MCP knowledge base. The information in the ontology provides classification for its resources based on type and function. The ontology allows a client application to obtain domain specific information about MCP

[4] Jini Architectural Overview, p 7

services. This information is necessary to discover MCP services provided by monitors without any *a priori* knowledge of the monitors themselves. A client can describe the type of resource they are interested in and the type of information about that resource that is of concern. Jini lookup services would provide the client application with the ability to describe the resources and information needs and then to identify all of the monitors that potentially can satisfy their specific information requirements. Following the MCP design concept, a client's information requirements can require the integration of information from several monitors.

When a client queries the Jini Look-Up service it receives a set of proxies for the monitors that can potentially satisfy the information requirements. The client application then can use this proxy to query the monitors to determine the best set of information sources. The client then uses the selected proxy(ies) to communicate directly to the monitor(s).

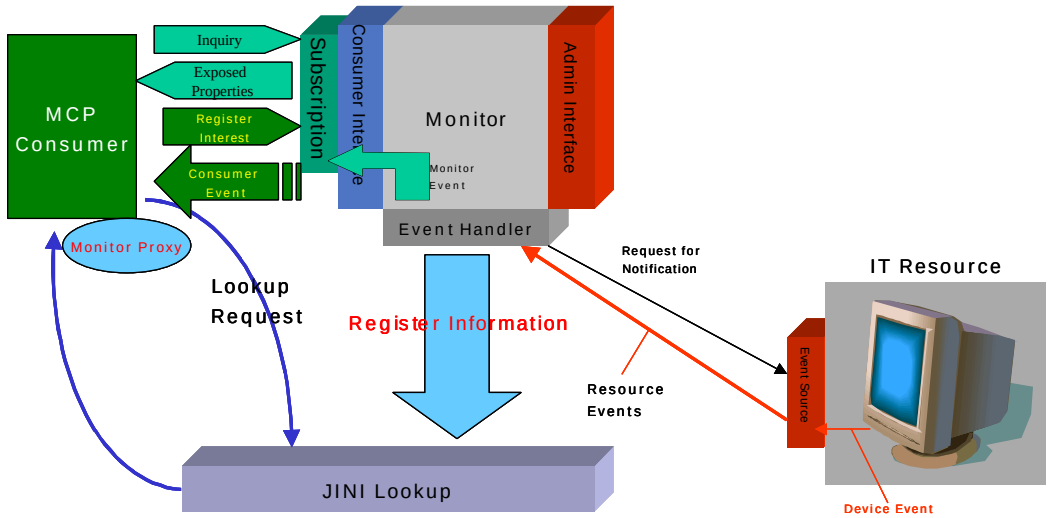

Figure 6: MCP Component Relationships

5.2.2 Contract-Based Service Delivery

As a result of the negotiation process, the client and the monitor service(s) enter into a service contract. The service contract specifies the client's data requirements, the data processing instructions for the monitor, and the duration of the contract. The duration is managed in the form of a lease. The client is responsible for renewing the lease in order to maintain the service. The contract is implemented through the use of a mobile code module that is owned by the client but operated by the service provider.

The mobile code elements provide specific processing instructions that are executed by the monitor. The monitor uses the mobile code to process and filter events that would be reported to the clients. These instructions will usually establish reporting thresholds or criteria to minimize the traffic between monitors and clients.

5.2.3 Service Leases

As stated above, the service contract specified a valid timeframe for the contract. This time, expressed in duration from the time of initialization, is referred to as a lease.

The leasing concept is used within Jini to help ensure the validity of Jini services in a distributed environment. Leasing provides a mechanism for service providers to terminate contracts that are no longer valid or to eliminate references to services that are no longer available in the environment. The client establishes and renews the lease with the service provider. The service provider is responsible for managing the current leases and discarding leases (and terminating contracts) when a lease is not renewed. The monitor establishes a lease with the Jini look-up service at registration. The monitor is responsible for renewing this lease in a timely manner. In turn, a client establishes a lease for services with the monitor. It is the client's responsibility to renew the lease. The monitor terminates service contracts for all invalid leases.

The service-leasing concept provides a self-healing capability to the distributed system. It is through this leasing mechanism that services and clients can discover invalid leases. Invalid leases are the results of a variety of occurrences in a distributed system. A client or service could have an invalid lease due to a network failure; a failure of a leasing service to renew the lease in question or the process itself has been terminated. Regardless of the cause, the leasing mechanism provides a way for the service providers to detect and fix inconsistencies thus obtaining the self-healing property.

6 MCP Knowledge Base

Thus far this paper has addressed the conceptual framework of the Master Caution Panel, and the support provided by the underlying Jini technology. It is MCP's knowledge base however that separates MCP from commercially available monitoring and management systems. Work has just started on the knowledge base therefore only a few of the planned capabilities are available at the current time. The MCP knowledge base is a Jini service providing knowledge capabilities to MCP consumers.

The knowledge base provides a mapping service that links Air Operations Center tasks to the specific resources that support those tasks. This mapping allows MCP users configure MCP clients to discover appropriate MCP monitor services using only knowledge of specific tasks. The MCP knowledge base extends this mapping capability by providing a mapping of specific IT resource problems to their operational impact. The knowledge base also provides clients with indicators for these identified problems and provides, on demand, the specific service contract to be used to determine relevant IT resource status.

The knowledge base also identifies a set of recommended courses of action in response to the currently defined resource related problems. In the future, BBN hopes to add intelligence to the

business logic monitors to allow them to correlate multiple alerts, determine the most probable cause of the problem and then intelligently recommend the best course of action to the decision maker.

In the future the MCP knowledge base will also include modeling and simulation services that will support the analysis of resource usage patterns and the subsequent detection of anomalies in those patterns. Identifying and monitoring patterns of resource usage may allow future versions of MCP to report task completion status.

7 MCP Execution Concepts

7.1 MCP Deployment

Several deployment options are available for MCP deployment within an AOC. The first option is to deploy MCP in a truly distributed manner with monitors residing on virtually every piece of hardware in the AOC. Using this deployment concept, WBEM, SNMP and Database monitors would reside on the machines they were monitoring. Latency monitors would be strategically deployed throughout the AOC to measure and sample latency across the various LAN segments. This deployment concept could adversely impact the performance of the monitored resources by placing additional processing burdens on the servers and workstations. This could violate principle nine for monitoring distributed systems in that the monitor may contribute to identified resource problems. On the positive side, this deployment concept significantly reduces the SNMP and WBEM traffic between the resource, its event source, and the monitor. This also places a burden on the system administrator to install Jini and MCP libraries and Java virtual machines on each piece of hardware.

The second deployment concept for MCP calls for the deployment of MCP monitors and Jini services on one to four dedicated service hosts within the AOC. The advantage of this deployment concept is that the monitors do not impact the resources being monitored. Additionally, management and evolution of monitors is easier because all components are located on a few service hosts. The disadvantage to this approach is that SNMP (or WBEM) traffic must flow across the network from the resources to the monitors. Preliminary tests within the MCP development lab have indicated that this traffic is minimal but further tests are required in actual AOC configurations.

8 The Future

Master Caution Panel development is scheduled through 2003 with transition funding expected to take MCP capabilities from operational prototype to full deployment in 2004 and 2005. Additionally, there are several interesting technical opportunities for the MCP team. One potential opportunity it to team with the Defense Advanced Research Projects Agency (DARPA)

to bring the services based technologies of MCP together with DARPA's Dynamic Assembly for System Adaptability, Dependability and Assurance (DASADA) program. The DASADA program has developed a set of probes and gauges along with enhanced architectural representations that may be used by MCP to dynamically deploy monitors in response to user specified needs. The DASADA architectural representations could also be used to configure MCP for a specific deployment option. DASADA also offers a set of deployable worklets that could be deployed as shared services to repair certain It related problems either with or without direct human intervention.

MCP represents one of the first steps toward developing a services based architecture for command and control. It also provides a framework for developing future capabilities in support of Network Centric Warfare.

9 Bibliography:

Core Jini, W. Keith Edwards, Prentice-Hall, Sun Microsystems Press, 2nd Edition, 2001. Note, the 2nd edition covers Jini 1.1, whereas the first edition does not.

Paving the Way for Transparent Application and Data Interchange, Wayne W. Eckerson and Anne Thomas Manes, November 1999. A whitepaper about metadata exchange prepared for Sun Microsystems, Inc., by the Patricia Seybold Group

Jini Architectural Overview, Technical White Paper, Sun Microsystems Technical White Paper, Palo Alto, CA 1999

Rio Architecture Overview, Sun Microsystems Technical White Paper, Palo Alto, CA, 2001

Master Caution Panel Concept of Operations, United States Air Force Command and Control Battle Lab, March, 2002

BBN Distributed Modeling Architecture (BDMA), BBN Technologies, White Paper, San Diego CA, 2001

Javadoc documentation for BDMA.

.*The Javasoft Homepage*.

Sun's Jini Homepage.

Jini.org, the official forum for Jini technology.